PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-209651
(43)Date of publication of application : 03.08.2001

(51)Int.Cl. G06F 17/30

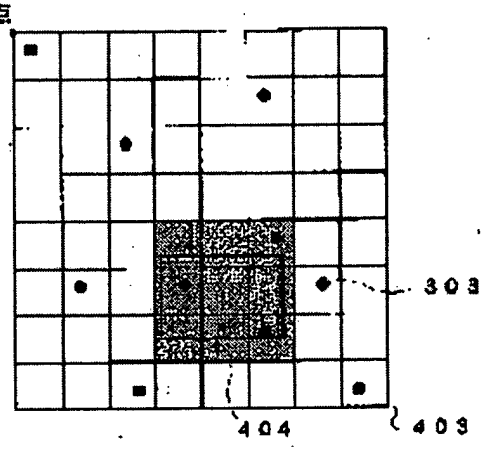
(21)Application number : 2000-017877
(22)Date of filing : 24.01.2000

(71)Applicant : HITACHI LTD
(72)Inventor : KAMIKAWA NOBUHIKO
IWASAKI KAZUMASA

(54) METHOD AND DEVICE FOR RETRIEVING MULTI-DIMENSIONAL VECTOR AND RECORDING MEDIUM HAVING MULTI-DIMENSIONAL VECTOR RETRIEVAL PROGRAM RECORDED THEREON

(57)Abstract:
PROBLEM TO BE SOLVED: To solve the problem that processing time and a processing amount required for the processing of retrieving n-dimensional vector data increase corresponding to the dimension number of vector data to be a retrieval object.
SOLUTION: In this multi-dimensional vector retrieval method for retrieving pertinent multi-dimensional vector data whose position and size are both present inside an optional multi-dimensional rectangular area inside a multi-dimensional space from the stored plural multi-dimensional vector data, the multi-dimensional rectangular area is inputted as a retrieval range, a first judgment processing is performed to a data pair composed of the multi-dimensional vector data and an address value calculated based on the rough values of the respective dimensions of the vector data by using the address value, a second judgment processing is performed by using the multi-dimensional vector data in the case that the address value is within a prescribed range and the multi-dimensional vector data of the data pair are outputted as a retrieved result in the case that the vector data are within the prescribed range.

図1



LEGAL STATUS

[Date of request for examination] 09.09.2003
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] In a multi-dimension vector search method which searches the multi-dimension vector data concerned to which a location and size exist in a multi-dimension rectangle field of arbitration out of two or more accumulated multi-dimension vector data As opposed to a data pair which consists of an address value which inputs said multi-dimension rectangle field as retrieval conditions, and is computed based on rough order of magnitude of each dimension of said multi-dimension vector data and the multi-dimension vector data concerned When 1st judgment processing is performed using the address value concerned and the address value concerned fulfills said retrieval conditions A multi-dimension vector search method characterized by outputting the multi-dimension vector data concerned as a retrieval result when 2nd judgment processing is performed using multi-dimension vector data of the data pair concerned and the vector data concerned fulfills said retrieval conditions.
[Claim 2] A multi-dimension vector search method characterized by associating said address value computed based on rough order of magnitude of each dimension of the multi-dimension vector data concerned, and the multi-dimension vector data concerned, and accumulating separately in a multi-dimension vector search method according to claim 1 in case said multi-dimension vector data is accumulated.
[Claim 3] A multi-dimension vector search method characterized by making into an address value a value which creates a vector for the addresses and is computed based on rough order of magnitude of each dimension of the vector for the addresses concerned by choosing a dimension of arbitration of said multi-dimension vector data in a multi-dimension vector search method according to claim 1 in case said address value is computed.
[Claim 4] In a multi-dimension vector search method according to claim 1, a regulation which attaches single dimension-sequence about said address value is set up. Create B-Tree which used said address value as an index key, and said data pair is accumulated. In case said 1st judgment processing is performed, use the B-Tree concerned and it asks for a storing location of the minimum address value of said retrieval range, and the maximum address value of said retrieval range. A multi-dimension vector search method characterized by not performing said 1st judgment processing to said data pair with an address value smaller than the minimum address value of the retrieval range concerned, and said data pair with a larger address value than the maximum address value of the retrieval range concerned.
[Claim 5] In a multi-dimension vector search method which searches the multi-dimension vector data concerned to which a location and size exist in a multi-dimension rectangle field of arbitration out of two or more accumulated multi-dimension vector data As opposed to a data pair which consists of the retrieval condition input section which inputs said multi-dimension rectangle field as retrieval conditions, and an address value computed based on rough order of magnitude of each dimension of said multi-dimension vector data and the multi-dimension vector data concerned The 1st judgment processing section which performs 1st judgment processing using the address value concerned, The 2nd judgment processing section which performs 2nd judgment processing using multi-dimension vector data of the data pair concerned when the address value concerned fulfills said retrieval conditions, Multi-dimension vector retrieval equipment characterized by having the retrieval result output section which outputs the multi-dimension vector data concerned as a retrieval result when the vector data concerned fulfills said retrieval conditions.
[Claim 6] In a multi-dimension vector search method which searches the multi-dimension vector data concerned to which a location and size exist in a multi-dimension rectangle field of arbitration out of two or more accumulated multi-dimension vector data As opposed to a data pair which consists of the retrieval condition input section which inputs said multi-dimension rectangle field as retrieval conditions, and an address value computed based on rough order of magnitude of each dimension of said multi-dimension vector data and the multi-dimension vector data concerned The 1st judgment processing section which performs 1st judgment processing using the address value concerned, The 2nd judgment processing section which performs 2nd judgment processing using multi-dimension vector data of the

data pair concerned when the address value concerned fulfills said retrieval conditions, Data medium characterized by recording a program for operating a computer as the retrieval result output section which outputs the multi-dimension vector data concerned as a retrieval result when the vector data concerned fulfills said retrieval conditions.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[The technical field to which invention belongs] This invention is applied to retrieval in case n becomes dozens or more from the data base with which much n (n>=1) dimension vector data was accumulated especially about the multi-dimension vector search method which searches the vector data to which a location and size exist in the n-dimensional rectangle field of arbitration in an n-space, and relates to effective technology.
[0002]
[Description of the Prior Art] Conventionally, the method of searching the vector data to which a location and size exist in the n-dimensional rectangle field of arbitration in n vector space is indicated by PCT/EP 97/04520 out of two or more accumulated n-dimensional vector data.
[0003] By this method, it divides into the field which can set vector space in order in single dimension, and vector data is managed by the field where each vector data belongs. At the time of retrieval, a location and size make the n-dimensional rectangle field of arbitration a retrieval range in vector space, it asks for all the fields that lap with a retrieval range, and the comparison with each dimension value of vector data, and the minimum value of each dimension of a retrieval range and maximum is performed to the vector data which exists in each field for which it asked. In the comparison of each dimension value of vector data, the vector data judged that exists in retrieval within the limits is outputted as a retrieval result.
[0004]
[Problem(s) to be Solved by the Invention] However, by the above conventional methods, when object data became a high order former vector-dimensional [dozens of / or more], there was a problem that the processing time taken to search increased. "Namely, the processing which asks for all the fields where retrieval processing laps with "retrieval range according to the above-mentioned well-known example" Each dimension value of vector data, and the minimum value of each dimension of a retrieval range, It can divide into the processing which performs the comparison with maximum". The processing order of "processing which asks for all the fields that lap with a retrieval range" It is O (2 nxa (n)) (throughput which judges [whether a laps with a retrieval range to one field, and] **). The processing order of "processing which performs the comparison with each dimension value of vector data, and the minimum value of each dimension of a retrieval range and maximum" is O (rxb (n)) (the number of data from which r brings a retrieval result, throughput to which b performs the comparison with each dimension of a retrieval range for every affair). For example, when object vector data is 32 dimensions, the processing order of "processing which asks for all the fields that lap with a retrieval range" is set to O (about 4 billionxa (n)). Even if it assumes the processing time of a to be 1 microsecond, "processing which asks for all the fields that lap with a retrieval range" will take 4000 seconds. Thus, by the above-mentioned conventional method, an increment of the number of dimension of object data increases the retrieval processing time. Although a multi-dimension vector search method is applied to a retrieval system like for example, similar image retrieval, it is thought that it is difficult for the retrieval processing time demanded in such a retrieval system to be "about several seconds" to the vector data of about 100,000 affairs, and to fill a demand with the above-mentioned conventional method.
[0005] In order to solve the above-mentioned problem, the purpose of this invention is to offer the technology of reducing the effect of the number of dimension of the vector data from which a location and size serve as a candidate for retrieval to the processing time taken to search the n-dimensional vector data which exists in the n-dimensional rectangle field of arbitration, even when n is dozens or more.
[0006]
[Means for Solving the Problem] In a multi-dimension vector search method which searches the multi-dimension

vector data concerned to which a location and size exist in a multi-dimension rectangle field of arbitration out of two or more accumulated multi-dimension vector data in order to solve the above-mentioned problem As opposed to a data pair which consists of an address value which inputs said multi-dimension rectangle field as retrieval conditions, and is computed based on rough order of magnitude of each dimension of said multi-dimension vector data and the multi-dimension vector data concerned When 1st judgment processing is performed using the address value concerned and the address value concerned fulfills said retrieval conditions When 2nd judgment processing is performed using multi-dimension vector data of the data pair concerned and the vector data concerned fulfills said retrieval conditions, it is made to output the multi-dimension vector data concerned as a retrieval result. That is, retrieval processing by this invention is divided into "the 2nd judgment processing" which asks for data which finally brings a retrieval result for data which serves as "the 1st judgment processing" which narrows down data which serves as a candidate of a retrieval result, and a candidate of a retrieval result. Processing order of the 1st judgment processing is O (Nxf (n)) (N is the number of are recording data, and f is judgment-every affair ago throughput). Processing order of the 2nd judgment processing Since it is O (N'xg (n)) (the number of data and g from which N' is set as the object of this judgment processing are this judgment throughput in every affair), processing order of the whole retrieval processing is set to O (Nxf (n) +N'xg (n)). Since an address value is computed based on rough order of magnitude of each dimension value of vector data, size of an address value becomes small and is set to f(n) <<g (n) from size of vector data as a result. Moreover, in this invention, it is expectable by performing 1st judgment processing to narrow down the number of data set as the object of the 2nd judgment processing to about 3 times of the number of data which brings a retrieval result with experiential observation. For example, by retrieval which outputs a retrieval result of 100 affairs, processing order of the 2nd judgment processing is set to O (300xg (n)) from are recording data of 100,000 affairs to the processing order O of the 1st judgment processing (100,000xf (n)). As the processing time in consideration of file I/O by experiential observation here For 10 microseconds, if the processing time of g is assumed to be 1 m seconds, since the processing time of about 1 second and the 2nd judgment processing will be about 0.3 seconds and the processing time of the 1st judgment processing will become 1.3 seconds by the whole retrieval processing about it, the processing time of f It can be said that it is the engine performance which may fill a demand of "about several seconds" to vector data of about 100,000 affairs. Moreover, a CPU operation hardly affects throughput of the whole retrieval processing, even if a number of dimension of vector data used as a candidate for retrieval increases, since the processing time is very short compared with file I/O.

[0007] Even when n is dozens or more as mentioned above according to the multi-dimension vector search method of this invention, it is possible to reduce even to time amount with which throughput does not increase the processing time taken to search n-dimensional vector data to which a location and size exist in a n-dimensional rectangle field of arbitration according to a number of dimension of vector data used as a candidate for retrieval, but a demand of "about several seconds" may be filled to vector data of about 100,000 affairs.

[0008]

[Embodiment of the Invention] Next, 1 operation gestalt of this invention is explained with reference to a drawing.

[0009] Drawing 1 is drawing showing the principle of the multi-dimension vector search method of this operation gestalt. With this operation gestalt, it narrows down to three vector data 303 which is the vector data 303 which may exist in the retrieval range 404 and which exists in the half-tone-dot-meshing portion of drawing from ten vector data 303 by performing 1st judgment processing first as a procedure of asking for the vector data 303 which exists in the retrieval range 404 from ten vector data 303 distributed over the multi-dimension space 403. Next, only from three narrowed-down vector data 303, 2nd judgment processing is performed and it asks for two vector data 303 which exists in the retrieval range 404. Since there is very little computational complexity of the 1st judgment processing here, the time amount which retrieval processing takes can be reduced.

[0010] Drawing 2 is drawing showing the configuration of the multi-dimension vector retrieval equipment of this operation gestalt. As shown in drawing 2 , the multi-dimension vector retrieval equipment of this operation gestalt consists of CPU100, an input unit 101, an output unit 102, a bus 103, memory 200, and a secondary storage 300. At the time of are recording processing activation of this system, the are recording processing program 201, the address calculation program 203, and the B-Tree program 204 are stored in memory 200, and it performs by CPU100. Over memory 200, the are recording processing program 201 updates the B-Tree data 301, and the vector data 303 inputted from the input unit 101 stores the address value 302 and vector data 303 in a secondary storage 300. At the time of retrieval processing activation of this system, the retrieval processing program 202, the address calculation program 203, the B-Tree program 204, the axial address calculation program 205, and the within the limits judging program 206 are stored in memory 200, and it performs by CPU100. Here, the B-Tree program 204 is a program which performs B-Tree processing generally used. Over memory 200, the retrieval processing program 202 asks for the vector data 303 of

a retrieval result with reference to the B-Tree data 301, the address value 302, and vector data 303, and the information on the retrieval range 404 that it was inputted from the input unit 101 outputs the vector data 303 of a retrieval result to an output unit 102. The B-Tree data 301, the address value 302, and vector data 303 are stored in the secondary storage 300. The B-Tree data 301 is data created by the B-Tree program 204, and as shown in drawing 10 , it is equivalent to the node section of B-Tree which used the address value 302 as the index key. It is vector data 303 that it is equivalent to the live-data section of the address value 302 and the B-Tree data 301 that it is equivalent to the leaf section of the B-Tree data 301. Hereafter, a group with the vector data 303 of the origin which computed the address value 302 and the address value 302 concerned is called a data pair.

[0011] the are recording processing program 201 for operating this system, the retrieval processing program 202, the address calculation program 203, the B-Tree program 204, the axial address calculation program 205, and the within the limits judging program 206 -- since -- after being recorded on record media, such as CD-ROM, and being stored in secondary-storage data medium 2, the becoming program shall be loaded to memory 12 and shall be performed In addition, other data medium other than CD-ROM is sufficient as data medium which records said program.

[0012] Hereafter, are recording processing and retrieval processing of this operation gestalt are explained.

[0013] Drawing 3 is the flow chart of the are recording processing program 201 of this operation gestalt. It combines with drawing 11 and drawing 12 hereafter, and explains. At step 31, the address value 302 is computed by the address calculation program 203 from the vector data 303 inputted from the input unit 101. If vector data 303 (11) is inputted as shown in drawing 11 , the address value 302 [1.3.2] of vector data 303 (11) will be computed. At step 32, it asks for the storing location in the secondary storage 300 of the computed address value 302 by the B-Tree program 204, the B-Tree data 301 is updated, and the address value 302 and vector data 303 are stored in a secondary storage 300. Here, since the size relation to data pair (3) < data pair (11) < data pair (4) of the address value 302 has become it is shown in drawing 12 -- as -- a data pair -- the address value 302 of (11) [1.3.2] -- a data pair, since it is inserted between the address values 302 [2.0.3] of (the address value 302 of 3) [1.0.3], and data pair (4) The B-Tree data 301 is updated and the address value 302 and vector data 303 are stored in an appropriate storing location in connection with it.

[0014] Drawing 4 is the flow chart of the address calculation program 203 of this operation gestalt. At step 41, the vector for the addresses is created by choosing m required dimensions according to the use of retrieval from the vector data 303 of the object which computes the address value 302. Let the cube 401 which consists of vector space for the addresses be a processing-object cube at step 42. "1" is substituted for Variable i at step 43. i is a variable which confirms whether it processed to all the digits of the address value 302. At step 44, radical duty rate processing is performed to an object cube, and a processing-object cube is divided into the 2m piece subcube 402. At step 45, vector data 303 judges in which subcube 402 it is contained among the subcubes 402 divided at step 44. Let the number of the subcube 402 in which vector data 303 is contained be the number Xi of the i-th figure of the address value 302 at step 46. Here, Xi is sign-less m binary integer. Let the subcube 402 in which vector data 303 is contained be a processing-object cube at step 47. At step 48, it is confirmed whether it processed to all the digits of the address value 302. Since it progressed to step 49 when i<k was filled, and it processed to all the digits of the address value 302 when not filling, a program is ended by making into the address value 302 the k numerical lists [X1.X2.--.Xk]. "1" increment of the variable i is carried out at step 49. Here, if the digit count k of the address value 302 is made [many], in the comparison processing of the address value 302 performed by the within the limits judging program 206, judgment precision becomes good, and the count which performs comparison processing of vector data 303 can be reduced. On the other hand, the computational complexity of the comparison processing [itself] of the address value 302 and the amount of data of the address value 302 become large. Therefore, it is necessary to set the digit count k of the address value 302 as a suitable value in consideration of the distribution density of the vector data 303 accumulated etc.

[0015] As a radical duty rate, as shown in drawing 8 , it is the processing divided into the 2m piece subcube 402 by dividing each side 2 about the m-dimensional cube 401, and the serial number is attached to each subcube 402. Here, although each side is divided the 2nd grade in the case of division, you may divide in consideration of the bias of distribution of vector data 303. With the-like 1-dimensional sequencing regulation of the address value 302, it is defined as large in about 302 address value with the large numeric value of a high-order digit. That is, when the address value 302 of triple figures is computed from the two-dimensional vector for the addresses, if the address value 302 has size relation like a formula 1 and this is put in order in ascending order, it will serve as sequence shown by the arrow head of drawing 9 .

[0016]

[Equation 1] 0.0.0< 0.0.1 -- < -- -- < -- 0.0.3< 0.1.0 -- < -- -- <3.3.2<3.3.3 drawing 5 is the flow chart of the retrieval processing program 202 of this operation gestalt. Hereafter, to are recording data when the retrieval range 404 of drawing 13 is inputted, the flow of the processing which a retrieval processing program performs is combined with

drawing 14 , and is explained. At step 50, the nearest point from a zero about in the retrieval range 404 and the furthest point are searched for, and the address value 302 of each point is computed by the address calculation program 203. The address value 302 in which the furthest point from a zero about in the retrieval minimum address value and the retrieval range 404 has the address value 302 which the nearest point from a zero about in the retrieval range 404 has hereafter is called the retrieval maximum address value. The [2.1.1] retrieval maximum address value is computed for the retrieval minimum address value with [3.2.1]. At step 51, it asks for the storing location of the retrieval minimum address value and the retrieval maximum address value in a secondary storage 300 by the B-Tree program 204. Since the storing location of the retrieval maximum address value is called for for the storing location of the retrieval minimum address value between (9) and (10) between (4) and (5) as shown in drawing 14 At step 52, the axial address value of the axial address value of the retrieval minimum address value and the retrieval maximum address value is computed by the axial address calculation program. The axial address value of the retrieval minimum address value is computed with a horizontal axis [0.1.1] and an axis of ordinate [1.0.0], and the axial address value of the retrieval maximum address value is computed with a horizontal axis [1.0.1] and an axis of ordinate [1.1.0]. At step 53, the data pair stored in the storing location of the retrieval minimum address value calculated at step 51 is made into a processing-object data pair. That is, data pair (5) is made into a processing-object data pair. At step 54, it judges whether it is necessary to process to the data pair stored after the processing-object data pair, i.e., a data pair with the larger address value 302 than the address value 302 of a processing-object data pair. This judgment is made by comparing the storing location of the retrieval maximum address value and the storing location of a processing-object data pair for which it asked at step 52. If it "storing location < processing-object data pair storing location of the retrieval maximum address value" Becomes, since it is not necessary to continue processing, a program is ended. If it "storing location >= processing-object data pair storing location of the retrieval maximum address value" Becomes, it will progress to step 55. That is, to data pair (10) stored after data pair (9), it is not necessary to perform judgment processing in drawing 14 . At step 55, the axial address of the address value 302 of a processing-object data pair is computed by the axial address calculation program 205. The axial address of the address value 302 (5) is called for with a horizontal axis [0.1.1] and an axis of ordinate [1.0.1]. At step 56, the within the limits judging program 206 performs judgment processing of whether the vector data 303 of a processing-object data pair exists in the retrieval range 404. At step 57, it would be judged with the vector data 303 of a processing-object data pair existing in the retrieval range 404, namely, if the vector data 303 of a processing-object data pair is as a result of retrieval, it will progress to step 58. If that is not right, it will progress to ******** 59. At step 58, it outputs to an output unit 102 by making the vector data 303 of a processing-object data pair into a retrieval result. At step 59, it progresses to step 54 by making into a processing object the data pair stored in the next location of a processing-object data pair. A processing-object data pair makes it (data pair (6 in which the processing-object data pair is stored by the degree when it is 5)).

[0017] as mentioned above -- B-Tree -- a program -- 204 -- using -- things -- data -- a pair -- (-- one --) -- data -- a pair -- (-- two --) -- data -- a pair -- (-- three --) -- data -- a pair -- (-- four --) -- data -- a pair -- (-- ten --) -- receiving -- the address -- a value -- 302 -- a comparison -- depending -- a judgment -- processing -- it is not necessary to carry out -- since -- the address --

[0018] Drawing 6 is the flow chart of the axial address calculation program 205 of this operation gestalt. j shaft address value aj is used in order to accelerate the processing which evaluates the address value 302 only about the j-th vector. "1" is substituted for Variable i at step 61. i is a variable which confirms whether the binary number was computed to all the digits of the address value 302. At step 62, each digit counts xi1, xi2, --, xim when displaying the number Xi of the i-th figure of the address value 302 with a binary number are computed. m is the number of dimension of the vector for the addresses here. At step 63, it is confirmed whether the binary number was computed to all digits. In filling i<k, it progresses to step 64, and in not filling, since the binary number was computed to all digits, it progresses to step 65. k is the number of dimension of the vector for the addresses here. "1" increment of the variable i is carried out at step 64. "1" is substituted for Variable j at step 65. j is a variable which confirms whether the axial address value in all the dimensions of the vector for the addresses was computed. At step 66, k binary integer which consists of x1j and a binary number of k pieces of x2 j, --, xkj among the binary numbers computed at step 62 is computed as a j shaft address value aj. At step 67, it is confirmed whether the axial address value in all the dimensions of the vector for the addresses was computed. In filling j<m, it progresses to step 68, and in not filling, since the axial address value in all the dimensions of the vector for the addresses was computed, it ends a program. "1" increment of the variable j is carried out at step 68.

[0019] Drawing 7 is the flow chart of the within the limits judging program 206 of this operation gestalt. Hereafter, the flow of the processing which the within the limits judging program 206 performs to the are recording data of drawing 1 is combined with drawing 15 and drawing 16 , and is explained. The processing which the within the limits judging

program 206 performs can be divided into "judgment processing by the comparison of the address value 302", and "judgment processing by the comparison of vector data 303 each dimension value." From step 71 to the step 74 is judgment processing by the comparison of the address value 302, and from step 75 to the step 78 is judgment processing by the comparison of vector data 303 each dimension value. "1" is substituted for Variable i at step 71. i is a variable which confirms whether judgment processing was performed to all the axial address values of the address value 302. At step 72, the address value 302 of a processing-object data pair judges whether it is a value from the retrieval minimum address value to the retrieval maximum address value. Here, i shaft address value of the address value 302 of a processing-object data pair [amini / ai / i shaft address value of the retrieval minimum address value and] and amaxi show i shaft address value of the retrieval maximum address value. In filling amini<=ai<=amaxi, it progresses to step 73. In not filling, the vector data 303 of a processing-object data pair ends a program noting that it does not exist in the interior of the retrieval range 404. At step 73, it is confirmed whether judgment processing was performed to all axial address values. m is the number of dimension of the vector for the addresses here. It progresses to step 75 noting that it progresses to step 74 in filling i<m, and the vector data 303 of a processing-object data pair exists in the interior of the retrieval range 404 since judgment processing was performed to all axial address values in not filling. "1" increment of the variable i is carried out at step 74. The detailed contents of the judgment processing by the comparison of the address value 302 are shown in drawing 15 . (5) The data judged that judgment processing by the comparison of the address value 302 is performed, and may exist in the retrieval range 404 to the data of - (9) is (5), (7), and (9). "1" is substituted for Variable j at step 75. j is a variable which confirms whether judgment processing was performed to all the dimensions of vector data 303. At step 76, the dimension value of the vector data 303 of a processing-object data pair judges whether it is a value from the maximum of the dimension of the retrieval range 404 concerned to the minimum value. Here, the j-dimensional value of the vector data 303 of a processing-object data pair [vminj / vj / the j-dimension minimum value of the retrieval range 404 and] and vmaxj show the j-dimension maximum of the retrieval range 404. In filling vminj<=vj<=vmaxj, it progresses to step 77. In not filling, the vector data 303 of a processing-object data pair ends a plug ram noting that it does not exist in the interior of the retrieval range 404. At step 77, it is confirmed whether judgment processing was performed to all the dimensions of vector data 303. n is the number of dimension of vector data 303 here. Since it progressed to step 78 when j<n was filled, and judgment processing was performed to all the dimensions of vector data 303 when not filling, the NEKUTORU data 303 of a processing-object data pair ends a program noting that it exists in the interior of the retrieval range 404. "1" increment of the variable j is carried out at step 78. The detailed contents of the judgment processing by the comparison of vector data 303 each dimension value are shown in drawing 16 . To the data of (5), (7), and (9), judgment processing by the comparison of the address value 302 is performed, the data judged that exists in the retrieval range 404 is (5) and (9), and two data of an individual brings a retrieval result.

[0020] With this operation gestalt, the image characteristic quantity of 590-dimensional one which analyzes brightness differential information and color information and is calculated from the image of one sheet was used as vector data 303. And the 32-dimensional vector for the addresses was created from the 590-dimensional vector data 303, and the address value 302 of 6 figures was computed. Therefore, the size of the address value 302 becomes 32/590, and the computational complexity which retrieval processing takes can be reduced rather than it computes the address value 302 from the 590-dimensional vector data 303. Moreover, the size of an axial address value is 6 bits, and the size of each dimension value of vector data 303 is 4 bytes = 32 bits (real number value) to it. By using the B-Tree program 204 to the set of the vector data 303 of 100,000 affairs here The vector data 303 set as the object of the judgment processing by the comparison of the address value 302 is narrowed down to one half of the are recording data number of cases. The retrieval processing which is judged as 300 affairs existing in the interior of the retrieval range 404 by the judgment processing by the comparison of the address value 302, and is judged as 100 affairs existing in the interior of the retrieval range 404 by the judgment processing by the comparison of vector data 303 each dimension value is considered. In this case, it can be said that the judgment processing by the comparison of the address value 302 for 50,000 affairs and the judgment processing by the comparison of vector data 303 each dimension value for 300 affairs were substituted for the judgment processing by the comparison of vector data 303 each dimension value for 100,000 affairs. Here, throughput using the B-Tree program 204 was taken as such a small thing that it can ignore compared with other processings. To the judgment processing by the comparison of vector data 303 each dimension value to one affair being 32 bit-comparison processing in 590-dimensional each dimension of vector data 303, since the judgment processing by the comparison of the address value 302 to one affair is 6 bit-comparison processing in 32-dimensional each dimension of the vector for the addresses, 32 / comparison processing-object data size per 590 or time becomes [the count of comparison processing per affair] 6/32. Therefore, if judgment processing by the comparison of vector data 303 each dimension value to one affair is set to 1u, it will become possible to reduce the computational complexity

which retrieval processing takes to abbreviation 500u+300from 100,000u u=about 800 u.
[0021] As explained above, according to the multi-dimension vector search method of this operation gestalt It is possible to reduce the processing time taken to search the n-dimensional vector data to which a location and size exist in the n-dimensional rectangle field of arbitration in n vector space about to 1/100. Moreover, since throughput does not increase according to the number of dimension of vector data used as the candidate for retrieval, it is possible to apply, even when n is dozens or more.
[0022]
[Effect of the Invention] In the processing which searches the n-dimensional vector data to which a location and size exist in the n-dimensional rectangle field of arbitration according to this invention By introducing "the 1st judgment processing" and "the 2nd judgment processing" Even when n is dozens or more, it becomes possible to reduce the effect of the number of dimension of the vector data from which a location and size serve as a candidate for retrieval to the processing time taken to search the n-dimensional vector data which exists in the n-dimensional rectangle field of arbitration.

[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－209651
（P2001－209651A）
(43)公開日　平成13年８月３日(2001.8.3)

(51)Int.Cl.'	識別記号	ＦＩ		テーマコード'（参考）
Ｇ０６Ｆ　17/30		Ｇ０６Ｆ　15/403	３４０Ｚ	５Ｂ０７５
		15/401	３２０Ｚ	

審査請求　未請求　請求項の数６　ＯＬ　（全 11 頁）

(21)出願番号　特願2000－17877(P2000－17877)

(22)出願日　平成12年１月24日(2000.1.24)

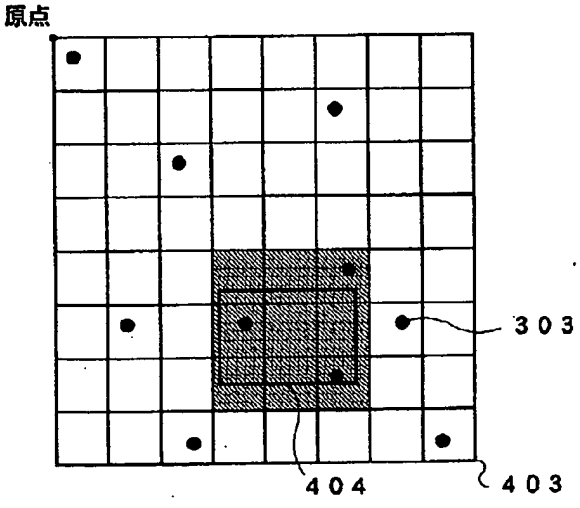
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　上川　伸彦
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所システム開発本部内
(72)発明者　岩崎　一正
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所システム開発本部内
(74)代理人　100075096
弁理士　作田　康夫

Ｆターム(参考)　5B075 NR20 PR10 QP05 QS20

(54)【発明の名称】　多次元ベクトル検索方法および装置並びに多次元ベクトル検索プログラムを記録した記録媒体

(57)【要約】
【課題】ｎ次元ベクトルデータを検索する処理に要する処理時間を、処理量が検索対象となるベクトルデータの次元数にしたがって増大するという問題がある。
【解決手段】蓄積された複数の多次元ベクトルデータの中から、多次元空間内において、位置、サイズ共に任意の多次元矩形領域内に存在する当該多次元ベクトルデータを検索する多次元ベクトル検索方法において、前記多次元矩形領域を検索範囲として入力し、前記多次元ベクトルデータと当該ベクトルデータの各次元の概略値を元に算出されるアドレス値とからなるデータ対に対して、当該アドレス値を用いて第１の判定処理を行い、当該アドレス値が所定範囲内の場合に、当該多次元ベクトルデータを用いて第２の判定処理を行い、当該ベクトルデータが所定範囲内の場合に、当該データ対の当該多次元ベクトルデータを検索結果として出力する。

図１



1

【特許請求の範囲】

【請求項１】蓄積された複数の多次元ベクトルデータの中から、位置、サイズ共に任意の多次元矩形領域内に存在する当該多次元ベクトルデータを検索する多次元ベクトル検索方法において、前記多次元矩形領域を検索条件として入力し、前記多次元ベクトルデータと当該多次元ベクトルデータの各次元の概略値を元に算出されるアドレス値とからなるデータ対に対して、当該アドレス値を用いて第１の判定処理を行い、当該アドレス値が前記検索条件を満たす場合に、当該データ対の多次元ベクトルデータを用いて第２の判定処理を行い、当該ベクトルデータが前記検索条件を満たす場合に、当該多次元ベクトルデータを検索結果として出力することを特徴とする多次元ベクトル検索方法。

【請求項２】請求項１記載の多次元ベクトル検索方法において、前記多次元ベクトルデータを蓄積する際に、当該多次元ベクトルデータの各次元の概略値を元に算出される前記アドレス値と当該多次元ベクトルデータとを関連付けて別々に蓄積しておくことを特徴とする多次元ベクトル検索方法。

【請求項３】請求項１記載の多次元ベクトル検索方法において、前記アドレス値を算出する際に、前記多次元ベクトルデータの任意の次元を選択する事によってアドレス用ベクトルを作成し、当該アドレス用ベクトルの各次元の概略値を元に算出される値をアドレス値とすることを特徴とする多次元ベクトル検索方法。

【請求項４】請求項１記載の多次元ベクトル検索方法において、前記アドレス値について一次元的順序を付ける規則を設定し、前記アドレス値をインデクスキーとしたＢ－Ｔｒｅｅを作成して前記データ対を蓄積し、前記第１の判定処理を行う際に、当該Ｂ－Ｔｒｅｅを用いて前記検索範囲の最小アドレス値と前記検索範囲の最大アドレス値の格納位置を求め、当該検索範囲の最小アドレス値よりも小さいアドレス値を持つ前記データ対と当該検索範囲の最大アドレス値よりも大きいアドレス値を持つ前記データ対に対しては、前記第１の判定処理を行わないことを特徴とする多次元ベクトル検索方法。

【請求項５】蓄積された複数の多次元ベクトルデータの中から、位置、サイズ共に任意の多次元矩形領域内に存在する当該多次元ベクトルデータを検索する多次元ベクトル検索方法において、前記多次元矩形領域を検索条件として入力する検索条件入力部と、前記多次元ベクトルデータと当該多次元ベクトルデータの各次元の概略値を元に算出されるアドレス値とからなるデータ対に対して、当該アドレス値を用いて第１の判定処理を行う第１の判定処理部と、当該アドレス値が前記検索条件を満たす場合に、当該データ対の多次元ベクトルデータを用いて第２の判定処理を行う第２の判定処理部と、当該ベクトルデータが前記検索条件を満たす場合に、当該多次元ベクトルデータを検索結果として出力する検索結果出力部とを備えることを特徴とする多次元ベクトル検索装置。

2

【請求項６】蓄積された複数の多次元ベクトルデータの中から、位置、サイズ共に任意の多次元矩形領域内に存在する当該多次元ベクトルデータを検索する多次元ベクトル検索方法において、前記多次元矩形領域を検索条件として入力する検索条件入力部と、前記多次元ベクトルデータと当該多次元ベクトルデータの各次元の概略値を元に算出されるアドレス値とからなるデータ対に対して、当該アドレス値を用いて第１の判定処理を行う第１の判定処理部と、当該アドレス値が前記検索条件を満たす場合に、当該データ対の多次元ベクトルデータを用いて第２の判定処理を行う第２の判定処理部と、当該ベクトルデータが前記検索条件を満たす場合に、当該多次元ベクトルデータを検索結果として出力する検索結果出力部としてコンピュータを機能させる為のプログラムを記録したことを特徴とする媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、多数のｎ（ｎ≧１）次元ベクトルデータが蓄積されたデータベース等から、ｎ次元空間内において、位置、サイズ共に任意のｎ次元矩形領域内に存在するベクトルデータを検索する多次元ベクトル検索方法に関し、特に、ｎが数十以上となる場合の検索に適用して有効な技術に関するものである。

【０００２】

【従来の技術】従来、蓄積された複数のｎ次元ベクトルデータの中から、ｎ次元ベクトル空間内において、位置、サイズ共に任意のｎ次元矩形領域内に存在するベクトルデータを検索する方法が、ＰＣＴ／ＥＰ９７／０４５２０に開示されている。

【０００３】この方法では、ベクトル空間を一次元的に順序付けることのできる領域に分割し、各ベクトルデータが属する領域によってベクトルデータを管理する。検索時には、ベクトル空間内で位置、サイズ共に任意のｎ次元矩形領域を検索範囲として、検索範囲と重なる領域を全て求め、求めた各領域内に存在するベクトルデータに対して、ベクトルデータの各次元値と検索範囲の各次元の最小値、最大値との比較を行う。ベクトルデータの各次元値の比較において、検索範囲内に存在すると判定されたベクトルデータを検索結果として出力する。

【０００４】

【発明が解決しようとする課題】しかしながら、上記のような従来の方法では、対象データが数十次元以上の高次元ベクトルになると、検索するのに要する処理時間が増大するという問題があった。すなわち、上記公知例によれば、検索処理は「検索範囲と重なる領域を全て求める処理」と「ベクトルデータの各次元値と検索範囲の各次元の最小値、最大値との比較を行う処理」とに分ける

ことができ、「検索範囲と重なる領域を全て求める処理」の処理オーダが、Ｏ（２ｎ×ａ（ｎ））（ａは１つの領域に対して検索範囲と重なるか否かをを判定する処理量）であり、「ベクトルデータの各次元値と検索範囲の各次元の最小値、最大値との比較を行う処理」の処理オーダが、Ｏ（ｒ×ｂ（ｎ））（ｒは検索結果となるデータ数、ｂは１件毎に検索範囲の各次元との比較を行う処理量）である。例えば、対象ベクトルデータが３２次元の場合には、「検索範囲と重なる領域を全て求める処理」の処理オーダは、Ｏ（約４０億×ａ（ｎ））となる。ａの処理時間を１μ秒と仮定しても、「検索範囲と重なる領域を全て求める処理」だけで４０００秒を要することになる。このように、上記従来の方法では、対象データの次元数が増加すると、検索処理時間が増大する。多次元ベクトル検索方法は、例えば類似画像検索のような検索システムに適用されるが、そのような検索システムにおいて要求される検索処理時間は１０万件程度のベクトルデータに対して「数秒程度」であり、上記従来の方法では、要求を満たすのは難しいと考えられる。

【０００５】本発明の目的は上記問題を改善するために、ｎが数十以上の場合でも、位置、サイズ共に任意のｎ次元矩形領域内に存在するｎ次元ベクトルデータを検索するのに要する処理時間に対し、検索対象となるベクトルデータの次元数の影響を減らす技術を提供することにある。

【０００６】

【課題を解決するための手段】上記問題を改善するために、蓄積された複数の多次元ベクトルデータの中から、位置、サイズ共に任意の多次元矩形領域内に存在する当該多次元ベクトルデータを検索する多次元ベクトル検索方法において、前記多次元矩形領域を検索条件として入力し、前記多次元ベクトルデータと当該多次元ベクトルデータの各次元の概略値を元に算出されるアドレス値とからなるデータ対に対して、当該アドレス値を用いて第１の判定処理を行い、当該アドレス値が前記検索条件を満たす場合に、当該データ対の多次元ベクトルデータを用いて第２の判定処理を行い、当該ベクトルデータが前記検索条件を満たす場合に、当該多次元ベクトルデータを検索結果として出力するようにする。すなわち、本発明による検索処理は、検索結果の候補となるデータを絞り込む「第１の判定処理」と、検索結果の候補となるデータを対象にして、最終的に検索結果となるデータを求める「第２の判定処理」とに分けられる。第１の判定処理の処理オーダは、Ｏ（Ｎ×ｆ（ｎ））（Ｎは蓄積データ数、ｆは１件毎の前判定処理量）であり、第２の判定処理の処理オーダは、Ｏ（Ｎ’×ｇ（ｎ））（Ｎ’は本判定処理の対象となるデータ数、ｇは１件毎の本判定処理量）であるので、検索処理全体の処理オーダはＯ（Ｎ×ｆ（ｎ）＋Ｎ’×ｇ（ｎ））となる。アドレス値はベクトルデータの各次元値の概略値を元に算出されるので、アドレス値のサイズはベクトルデータのサイズよりも小さくなり、結果として、ｆ（ｎ）＜＜ｇ（ｎ）となる。また、本発明においては、経験的観測により、第１の判定処理を行うことにより、第２の判定処理の対象となるデータ数を検索結果となるデータ数の３倍程度にまで絞り込むことが期待できる。例えば、１０万件の蓄積データから１００件の検索結果を出力する検索では、第１の判定処理の処理オーダＯ（１０万×ｆ（ｎ））に対して、第２の判定処理の処理オーダはＯ（３００×ｇ（ｎ））となる。ここで、経験的観測により、ファイルＩ／Ｏを考慮した処理時間として、ｆの処理時間を１０μ秒、ｇの処理時間を１ｍ秒と仮定すると、第１の判定処理の処理時間が１秒程度、第２の判定処理の処理時間が０．３秒程度であり、検索処理全体で１．３秒となるので、１０万件程度のベクトルデータに対して「数秒程度」という要求を満たし得る性能であるということができる。また、ＣＰＵ演算はファイルＩ／Ｏに比べて処理時間が非常に短いので、検索対象となるベクトルデータの次元数が増加しても、検索処理全体の処理量にはほとんど影響を与えない。

【０００７】以上のように本発明の多次元ベクトル検索方法によれば、ｎが数十以上の場合でも、位置、サイズ共に任意のｎ次元矩形領域内に存在するｎ次元ベクトルデータを検索するのに要する処理時間を、処理量が検索対象となるベクトルデータの次元数にしたがって増大せず、１０万件程度のベクトルデータに対して「数秒程度」という要求を満し得る時間にまで減らすことが可能である。

【０００８】

【発明の実施の形態】次に、本発明の一実施形態について図面を参照して説明する。

【０００９】図１は本実施形態の多次元ベクトル検索方法の原理を示す図である。本実施形態では、多次元空間４０３に分布する、１０個のベクトルデータ３０３から検索範囲４０４内に存在するベクトルデータ３０３を求める手順として、まず、第１の判定処理を行うことにより、１０個のベクトルデータ３０３から、検索範囲４０４内に存在し得るベクトルデータ３０３である、図の網掛け部分に存在する３個のベクトルデータ３０３に絞り込む。次に、絞り込まれた３個のベクトルデータ３０３に対してのみ、第２の判定処理を行い、検索範囲４０４内に存在する２個のベクトルデータ３０３を求める。ここで、第１の判定処理の計算量が非常に少ないので、検索処理に要する時間を減らすことができる。

【００１０】図２は本実施形態の多次元ベクトル検索装置の構成を示す図である。図２に示すように本実施形態の多次元ベクトル検索装置は、ＣＰＵ１００と、入力装置１０１と、出力装置１０２と、バス１０３と、メモリ２００と、二次記憶装置３００とから構成される。本システムの蓄積処理実行時は、蓄積処理プログラム２０１

と、アドレス算出プログラム２０３と、Ｂ－Ｔｒｅｅプログラム２０４とがメモリ２００に格納され、ＣＰＵ１００で実行される。入力装置１０１から入力されたベクトルデータ３０３がメモリ２００に渡り、蓄積処理プログラム２０１がＢ－Ｔｒｅｅデータ３０１を更新し、アドレス値３０２、ベクトルデータ３０３を二次記憶装置３００に格納する。本システムの検索処理実行時は、検索処理プログラム２０２と、アドレス算出プログラム２０３と、Ｂ－Ｔｒｅｅプログラム２０４と、軸アドレス算出プログラム２０５と、範囲内判定プログラム２０６とがメモリ２００に格納され、ＣＰＵ１００で実行される。ここで、Ｂ－Ｔｒｅｅプログラム２０４は、一般的に用いられているＢ－Ｔｒｅｅ処理を行うプログラムである。入力装置１０１から入力された検索範囲４０４の情報がメモリ２００に渡り、検索処理プログラム２０２がＢ－Ｔｒｅｅデータ３０１、アドレス値３０２、ベクトルデータ３０３を参照して検索結果のベクトルデータ３０３を求め、出力装置１０２に検索結果のベクトルデータ３０３を出力する。二次記憶装置３００には、Ｂ－Ｔｒｅｅデータ３０１と、アドレス値３０２と、ベクトルデータ３０３とが格納されている。Ｂ－Ｔｒｅｅデータ３０１は、Ｂ－Ｔｒｅｅプログラム２０４により作成されるデータであり、図１０に示すように、アドレス値３０２をインデクスキーとしたＢ－Ｔｒｅｅのノード部に相当する。Ｂ－Ｔｒｅｅデータ３０１のリーフ部に相当するのがアドレス値３０２、Ｂ－Ｔｒｅｅデータ３０１の実データ部に相当するのがベクトルデータ３０３となっている。以下、アドレス値３０２と当該アドレス値３０２を算出した元のベクトルデータ３０３との組をデータ対と呼ぶ。

【００１１】本システムを機能させるための、蓄積処理プログラム２０１、検索処理プログラム２０２、アドレス算出プログラム２０３、Ｂ－Ｔｒｅｅプログラム２０４、軸アドレス算出プログラム２０５、範囲内判定プログラム２０６、からなるプログラムは、ＣＤ－ＲＯＭ等の記録媒体に記録され二次記憶媒体２に格納された後、メモリ１２にロードされて実行されるものとする。なお前記プログラムを記録する媒体はＣＤ－ＲＯＭ以外の他の媒体でも良い。

【００１２】以下、本実施形態の蓄積処理と検索処理について説明する。

【００１３】図３は、本実施形態の蓄積処理プログラム２０１のフローチャートである。以下、図１１、図１２と併せて説明する。ステップ３１では、入力装置１０１から入力されたベクトルデータ３０３から、アドレス算出プログラム２０３によりアドレス値３０２を算出する。図１１に示すように、ベクトルデータ３０３（１１）が入力されたら、ベクトルデータ３０３（１１）のアドレス値３０２［１．３．２］を算出する。ステップ３２では、算出したアドレス値３０２の二次記憶装置３００における格納位置をＢ－Ｔｒｅｅプログラム２０４により求め、Ｂ－Ｔｒｅｅデータ３０１を更新し、アドレス値３０２、ベクトルデータ３０３を二次記憶装置３００に格納する。ここでは、アドレス値３０２の大小関係が、データ対（３）＜データ対（１１）＜データ対（４）となっているので、図１２に示すように、データ対（１１）のアドレス値３０２［１．３．２］はデータ対（３）のアドレス値３０２［１．０．３］と、データ対（４）のアドレス値３０２［２．０．３］との間に挿入されるので、Ｂ－Ｔｒｅｅデータ３０１を更新し、それに伴いアドレス値３０２、ベクトルデータ３０３をしかるべき格納位置に格納する。

【００１４】図４は、本実施形態のアドレス算出プログラム２０３のフローチャートである。ステップ４１では、アドレス値３０２を算出する対象のベクトルデータ３０３から、検索の用途に応じて必要なｍ個の次元を選択することにより、アドレス用ベクトルを作成する。ステップ４２では、アドレス用ベクトル空間からなるキューブ４０１を処理対象キューブとする。ステップ４３では、変数ｉに「１」を代入する。ｉとは、アドレス値３０２の全ての桁に対して処理を行ったかどうかをチェックする変数である。ステップ４４では、対象キューブに対して基本分割処理を行い、処理対象キューブを２ｍ個のサブキューブ４０２に分割する。ステップ４５では、ベクトルデータ３０３がステップ４４で分割されたサブキューブ４０２のうち、どのサブキューブ４０２に含まれるかを判断する。ステップ４６では、ベクトルデータ３０３が含まれるサブキューブ４０２の番号をアドレス値３０２のｉ桁目の数Ｘｉとする。ここで、Ｘｉは符号なしｍビット整数である。ステップ４７では、ベクトルデータ３０３が含まれるサブキューブ４０２を処理対象キューブとする。ステップ４８では、アドレス値３０２の全ての桁に対して処理を行ったかどうかをチェックする。、ｉ＜ｋを満たす場合にはステップ４９に進み、満たさない場合には、アドレス値３０２の全ての桁に対して処理を行ったので、［Ｘ１．Ｘ２．…．Ｘｋ］というｋ個の数値並びをアドレス値３０２としてプログラムを終了する。ステップ４９では、変数ｉを「１」増分する。ここで、アドレス値３０２の桁数ｋを多くすると、範囲内判定プログラム２０６で行われるアドレス値３０２の比較処理において判定精度が良くなり、ベクトルデータ３０３の比較処理を行う回数を減らすことができる。反面、アドレス値３０２の比較処理自体の計算量、アドレス値３０２のデータ量が大きくなる。そのため、アドレス値３０２の桁数ｋは、蓄積されているベクトルデータ３０３の分布密度等を考慮して適切な値に設定する必要がある。

【００１５】基本分割とは、図８に示すように、ｍ次元のキューブ４０１を各辺２分割することにより、２ｍ個のサブキューブ４０２に分割する処理であり、各サブキ

ューブ４０２には通し番号を付ける。ここでは、分割の際に各辺を２等分割しているが、ベクトルデータ３０３の分布の偏りを考慮して分割しても良い。アドレス値３０２の１次元的な順序付け規則とは、上位桁の数値が大きいアドレス値３０２ほど大きいと定義する。すなわち、２次元のアドレス用ベクトルから３桁のアドレス値３０２を算出した場合には、アドレス値３０２は数式１のような大小関係を持ち、これを昇順で並べると、図９の矢印で示した順番となる。

【００１６】

【数１】０.０.０＜０.０.１＜　…　＜０.０.３＜０.１.０＜　…　＜３.３.２＜３.３.３

図５は、本実施形態の検索処理プログラム２０２のフローチャートである。以下、図１３の検索範囲４０４が入力された場合の蓄積データに対して、検索処理プログラムが行う処理の流れを、図１４と併せて説明する。ステップ５０では、検索範囲４０４内で原点から一番近い点と一番遠い点とを求め、アドレス算出プログラム２０３により、それぞれの点のアドレス値３０２を算出する。以下、検索範囲４０４内で原点から一番近い点が持つアドレス値３０２を検索最小アドレス値、検索範囲４０４内で原点から一番遠い点が持つアドレス値３０２を検索最大アドレス値と呼ぶ。検索最小アドレス値は［２．１．１］、検索最大アドレス値は［３．２．１］と算出される。ステップ５１では、Ｂ－Ｔｒｅｅプログラム２０４により、二次記憶装置３００での、検索最小アドレス値と検索最大アドレス値の格納位置を求める。図１４に示すように、検索最小アドレス値の格納位置は（４）と（５）の間、検索最大アドレス値の格納位置は（９）と（１０）の間と求められるので、。ステップ５２では、軸アドレス算出プログラムにより、検索最小アドレス値の軸アドレス値と検索最大アドレス値の軸アドレス値を算出する。検索最小アドレス値の軸アドレス値は、横軸［０．１．１］、縦軸［１．０．０］と算出され、検索最大アドレス値の軸アドレス値は、横軸［１．０．１］、縦軸［１．１．０］と算出される。ステップ５３では、ステップ５１で求めた検索最小アドレス値の格納位置に格納されているデータ対を処理対象データ対とする。すなわち、データ対（５）を処理対象データ対とする。ステップ５４では、処理対象データ対以降に格納されているデータ対、すなわち、処理対象データ対のアドレス値３０２よりも大きいアドレス値３０２を持つデータ対に対して処理を行う必要があるかを判断する。この判断は、ステップ５２で求めた検索最大アドレス値の格納位置と処理対象データ対の格納位置とを比較することにより行う。「検索最大アドレス値の格納位置＜処理対象データ対格納位置」ならば、処理を続ける必要がないので、プログラムを終了する。「検索最大アドレス値の格納位置≧処理対象データ対格納位置」ならば、ステップ５５に進む。つまり、図１４において、データ対（９）よりも後に格納されているデータ対（１０）に対しては、判定処理を行う必要がない。ステップ５５では、軸アドレス算出プログラム２０５により、処理対象データ対のアドレス値３０２の軸アドレスを算出する。アドレス値３０２（５）の軸アドレスは、横軸［０．１．１］、縦軸［１．０．１］と求められる。ステップ５６では、範囲内判定プログラム２０６により、処理対象データ対のベクトルデータ３０３が検索範囲４０４内に存在するか否かの判定処理を行う。ステップ５７では、処理対象データ対のベクトルデータ３０３が検索範囲４０４内に存在すると判定された、すなわち、処理対象データ対のベクトルデータ３０３が検索結果であるならば、ステップ５８に進む。そうでないならば、すてっぷ５９に進む。ステップ５８では、処理対象データ対のベクトルデータ３０３を検索結果として出力装置１０２に出力する。ステップ５９では、処理対象データ対の次の位置に格納されているデータ対を処理対象として、ステップ５４に進む。処理対象データ対が（５）の場合、処理対象データ対を次に格納されているデータ対（６）にする。

【００１７】以上のように、Ｂ－Ｔｒｅｅプログラム２０４を用いることにより、データ対（１）、データ対（２）、データ対（３）、データ対（４）、データ対（１０）に対しては、アドレス値３０２の比較による判定処理を行う必要がないので、アドレス値３０２の比較による判定処理の計算量を１／２に減らすことができる。

【００１８】図６は、本実施形態の軸アドレス算出プログラム２０５のフローチャートである。ｊ軸アドレス値ａｊは、アドレス値３０２を第ｊベクトルについてのみ評価する処理を高速化するために使用される。ステップ６１では、変数ｉに「１」を代入する。ｉとは、アドレス値３０２の全ての桁に対して２進数を算出したかどうかをチェックする変数である。ステップ６２では、アドレス値３０２のｉ桁目の数Ｘｉを２進数で表示した時の各桁数ｘｉ１、ｘｉ２、…、ｘｉｍを算出する。ここでｍとは、アドレス用ベクトルの次元数である。ステップ６３では、全ての桁に対して２進数を算出したかどうかをチェックする。ｉ＜ｋを満たす場合にはステップ６４に進み、満たさない場合には、全ての桁に対して２進数を算出したのでステップ６５に進む。ここでｋとは、アドレス用ベクトルの次元数である。ステップ６４では、変数ｉを「１」増分する。ステップ６５では、変数ｊに「１」を代入する。ｊとは、アドレス用ベクトルの全ての次元における軸アドレス値を算出したかどうかをチェックする変数である。ステップ６６では、ステップ６２で算出された２進数のうちｘ１ｊ、ｘ２ｊ、…、ｘｋｊのｋ個の２進数から構成されるｋビット整数をｊ軸アドレス値ａｊとして算出する。ステップ６７では、アドレス用ベクトルの全ての次元における軸アドレス値を算出

したかどうかをチェックする。ｊ＜ｍを満たす場合にはステップ６８に進み、満たさない場合には、アドレス用ベクトルの全ての次元における軸アドレス値を算出したのでプログラムを終了する。ステップ６８では、変数ｊを「１」増分する。

【００１９】図７は、本実施形態の範囲内判定プログラム２０６のフローチャートである。以下、図１の蓄積データに対して範囲内判定プログラム２０６が行う処理の流れを、図１５、図１６と併せて説明する。範囲内判定プログラム２０６が行う処理は、「アドレス値３０２の比較による判定処理」と「ベクトルデータ３０３各次元値の比較による判定処理」とに分けることができる。ステップ７１からステップ７４までがアドレス値３０２の比較による判定処理であり、ステップ７５からステップ７８までがベクトルデータ３０３各次元値の比較による判定処理である。ステップ７１では、変数ｉに「１」を代入する。ｉとは、アドレス値３０２の全ての軸アドレス値に対して判定処理を行ったかどうかをチェックする変数である。ステップ７２では、処理対象データ対のアドレス値３０２が検索最小アドレス値から検索最大アドレス値までの値かどうかを判断する。ここで、ａｍｉｎｉとは検索最小アドレス値のｉ軸アドレス値、ａｉとは処理対象データ対のアドレス値３０２のｉ軸アドレス値、ａｍａｘｉとは検索最大アドレス値のｉ軸アドレス値を示す。ａｍｉｎｉ≦ａｉ≦ａｍａｘｉを満たす場合にはステップ７３へ進む。満たさない場合には、処理対象データ対のベクトルデータ３０３は検索範囲４０４の内部には存在しないとしてプログラムを終了する。ステップ７３では、全ての軸アドレス値に対して判定処理を行ったかどうかをチェックする。ここでｍとは、アドレス用ベクトルの次元数である。ｉ＜ｍを満たす場合にはステップ７４に進み、満たさない場合には、全ての軸アドレス値に対して判定処理を行ったので、処理対象データ対のベクトルデータ３０３は検索範囲４０４の内部に存在するとして、ステップ７５に進む。ステップ７４では、変数ｉを「１」増分する。図１５に、アドレス値３０２の比較による判定処理の詳細な内容を示す。（５）～（９）のデータに対して、アドレス値３０２の比較による判定処理を行い、検索範囲４０４内に存在し得ると判定されたデータが（５）、（７）、（９）である。ステップ７５では、変数ｊに「１」を代入する。ｊとは、ベクトルデータ３０３の全ての次元に対して判定処理を行ったかどうかをチェックする変数である。ステップ７６では、処理対象データ対のベクトルデータ３０３の次元値が検索範囲４０４の当該次元の最大値から最小値までの値かどうかを判断する。ここで、ｖｍｉｎｊとは検索範囲４０４のｊ次元での最小値、ｖｊとは処理対象データ対のベクトルデータ３０３のｊ次元値、ｖｍａｘｊとは検索範囲４０４のｊ次元での最大値を示す。ｖｍｉｎｊ≦ｖｊ≦ｖｍａｘｊを満たす場合にはステップ７７へ進む。満たさない場合には、処理対象データ対のベクトルデータ３０３は検索範囲４０４の内部には存在しないとしてプログラムを終了する。ステップ７７では、ベクトルデータ３０３の全ての次元に対して判定処理を行ったかどうかをチェックする。ここでｎとは、ベクトルデータ３０３の次元数である。ｊ＜ｎを満たす場合にはステップ７８に進み、満たさない場合には、ベクトルデータ３０３の全ての次元に対して判定処理を行ったので、処理対象データ対のネクトルデータ３０３は検索範囲４０４の内部に存在するとしてプログラムを終了する。ステップ７８では、変数ｊを「１」増分する。図１６に、ベクトルデータ３０３各次元値の比較による判定処理の詳細な内容を示す。（５）、（７）、（９）のデータに対して、アドレス値３０２の比較による判定処理を行い、検索範囲４０４内に存在すると判定されたデータが（５）、（９）であり、個の２個のデータが検索結果となる。

【００２０】本実施形態では、１枚の画像から輝度微分情報や色情報を解析して求められる５９０次元の画像特徴量をベクトルデータ３０３として使用した。そして、５９０次元のベクトルデータ３０３から３２次元のアドレス用ベクトルを作成し、６桁のアドレス値３０２を算出した。そのため、５９０次元のベクトルデータ３０３からアドレス値３０２を算出するよりも、アドレス値３０２のサイズが３２／５９０になり、検索処理に要する計算量を減らすことができる。また、軸アドレス値のサイズが６ビットであり、それに対して、ベクトルデータ３０３の各次元値のサイズは４バイト＝３２ビット（実数値）である。ここで、１０万件のベクトルデータ３０３の集合に対して、Ｂ－Ｔｒｅｅプログラム２０４を用いることによって、アドレス値３０２の比較による判定処理の対象となるベクトルデータ３０３を蓄積データ件数の１／２に絞り込み、アドレス値３０２の比較による判定処理で３００件が検索範囲４０４の内部に存在し得ると判定され、ベクトルデータ３０３各次元値の比較による判定処理で１００件が検索範囲４０４の内部に存在すると判定される検索処理について考察する。この場合、１０万件分のベクトルデータ３０３各次元値の比較による判定処理を５万件分のアドレス値３０２の比較による判定処理と３００件分のベクトルデータ３０３各次元値の比較による判定処理で代用したと言うことができる。ここで、Ｂ－Ｔｒｅｅプログラム２０４を用いた処理量は他の処理に比べて無視できるほど小さいものとした。１件に対するベクトルデータ３０３各次元値の比較による判定処理が、ベクトルデータ３０３の５９０次元各次元における３２ビット比較処理であるのに対して、１件に対するアドレス値３０２の比較による判定処理は、アドレス用ベクトルの３２次元各次元における６ビット比較処理であるので、１件当たりの比較処理回数が３２／５９０、１回あたりの比較処理対象データサイズ

11　　12

が６／３２となる。よって、１件に対するベクトルデータ３０３各次元値の比較による判定処理を１ｕとすると、検索処理に要する計算量を１０万ｕから、約５００ｕ＋３００ｕ＝約８００ｕに減らすことが可能となる。

【００２１】以上説明したように本実施形態の多次元ベクトル検索方法によれば、ｎ次元ベクトル空間内において、位置、サイズ共に任意のｎ次元矩形領域内に存在するｎ次元ベクトルデータを検索するのに要する処理時間を１／１００程度に減らすことが可能であり、また、処理量が検索対象となるベクトルデータの次元数にしたがって増大しないので、ｎが数十以上の場合でも適用することが可能である。

【００２２】

【発明の効果】本発明によれば、位置、サイズ共に任意のｎ次元矩形領域内に存在するｎ次元ベクトルデータを検索する処理において、「第１の判定処理」と「第２の判定処理」とを導入することによって、ｎが数十以上の場合でも、位置、サイズ共に任意のｎ次元矩形領域内に存在するｎ次元ベクトルデータを検索するのに要する処理時間に対し、検索対象となるベクトルデータの次元数の影響を減らすことが可能となる。

【図面の簡単な説明】

【図１】本実施形態の多次元ベクトル検索方法の原理を示す図である。

【図２】本実施形態の多次元ベクトル検索装置の構成を示す図である。

【図３】本実施形態の蓄積処理プログラム２０１のフローチャートである。

【図４】本実施形態のアドレス算出プログラム２０３のフローチャートである。

【図５】本実施形態の検索処理プログラム２０２のフローチャートである。

【図６】本実施形態の軸アドレス算出プログラム２０５のフローチャートである。

【図７】本実施形態の範囲内判定プログラム２０６のフローチャートである。

【図８】本実施形態のアドレス算出プログラム２０３が＊行う基本分割を説明する図である。

【図９】本実施形態のアドレス値３０２の１次元的な順序付け規則を説明する図である。

【図１０】本実施形態の二次記憶装置３００の詳細を示す図である。

【図１１】本実施形態の蓄積処理を説明するための多次元ベクトル空間４０３を示す図である。

【図１２】本実施形態の蓄積処理を説明するための二次記憶装置３００の詳細を示す図である。

【図１３】本実施形態の検索処理を説明するための多次元ベクトル空間４０３を示す図である。

【図１４】本実施形態の検索処理を説明するための二次記憶装置３００の詳細を示す図である。

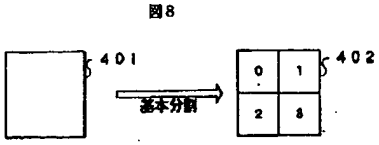
【図１５】本実施形態のアドレス値３０２の比較による判定処理の詳細な内容を示す図である。

【図１６】本実施形態のベクトルデータ３０３各次元値の比較による判定処理の詳細な内容を示す図である。

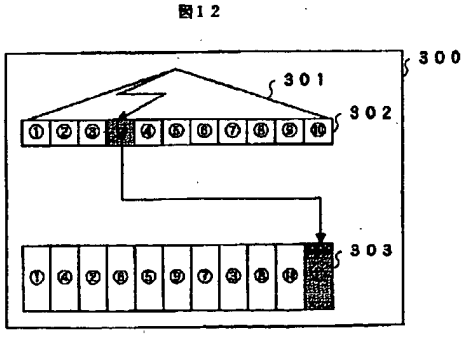
【符号の説明】

１０１　ＣＰＵ
１０１　入力装置
１０２　出力装置
１０３　バス
２００　メモリ
２０１　蓄積処理プログラム
２０２　検索処理プログラム
２０３　アドレス算出プログラム
２０４　Ｂ－Ｔｒｅｅプログラム
２０５　軸アドレス算出プログラム
２０６　範囲内判定プログラム
３００　二次記憶装置
３０１　Ｂ－Ｔｒｅｅデータ
３０２　アドレス値
３０３　ベクトルデータ
４０１　キューブ
４０２　サブキューブ
４０３　多次元ベクトル空間

【図８】

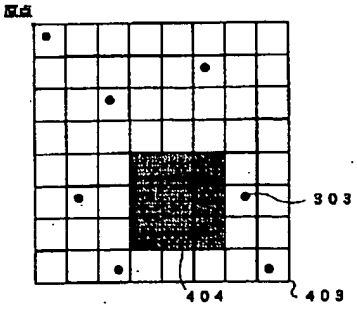


【図１２】



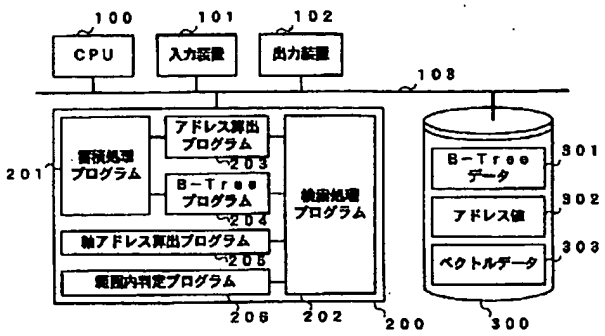
【図１】

図1



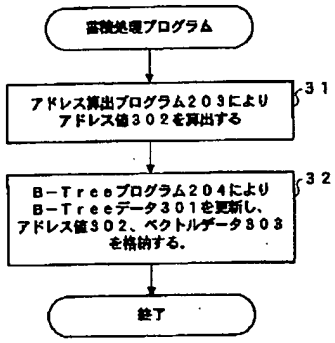
【図２】

図2



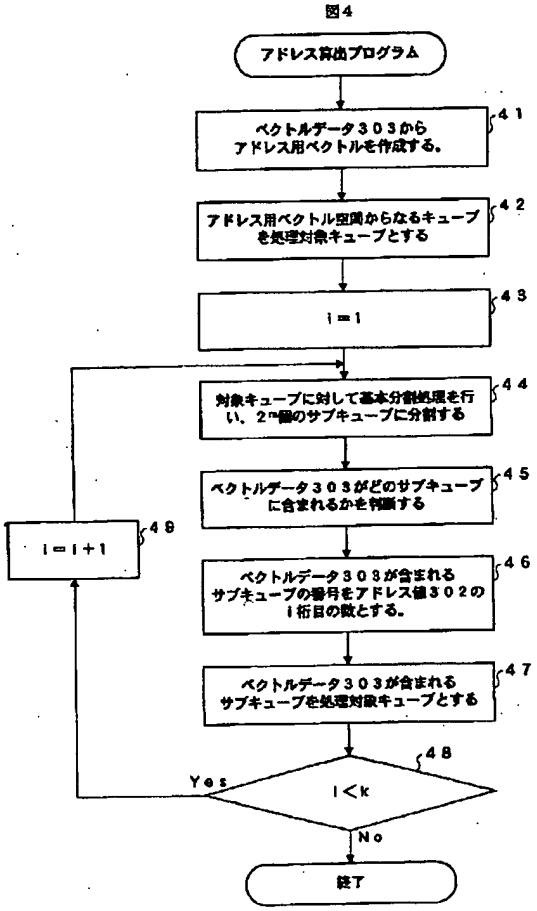
【図３】

図3



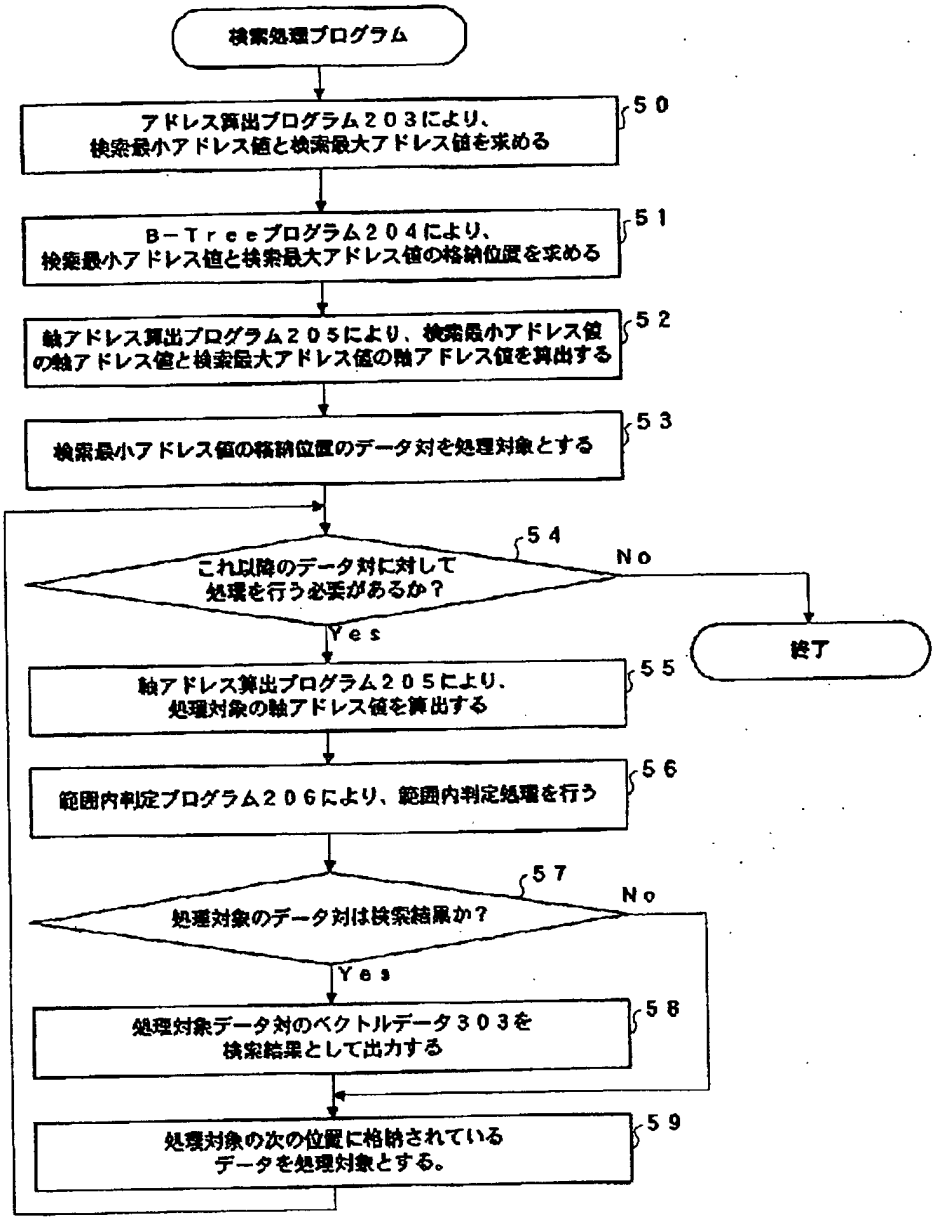
【図４】

図4



【図５】

図５



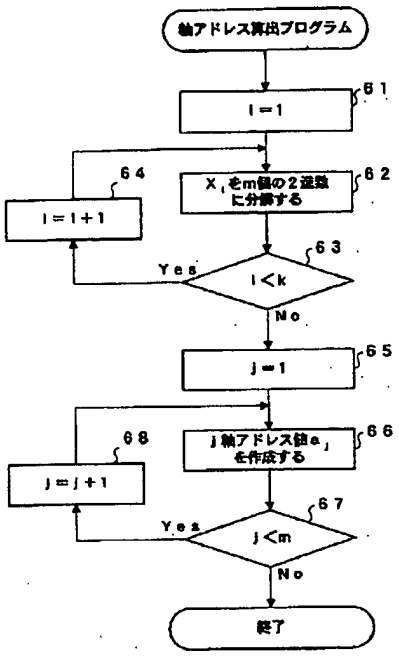
【図１６】

図１６

対象データ		判定条件式		判定結果
⑤	横軸	0.35 ≦ 0.4 ≦ 0.75	○	検索範囲内
	縦軸	0.6 ≦ 0.7 ≦ 0.85	○	
⑦	横軸	0.35 ≦ 0.7 ≦ 0.75	○	検索範囲外
	縦軸	0.6 ≦ 0.55 ≦ 0.85	×	
⑨	横軸	0.35 ≦ 0.7 ≦ 0.75	○	検索範囲内
	縦軸	0.6 ≦ 0.8 ≦ 0.85	○	

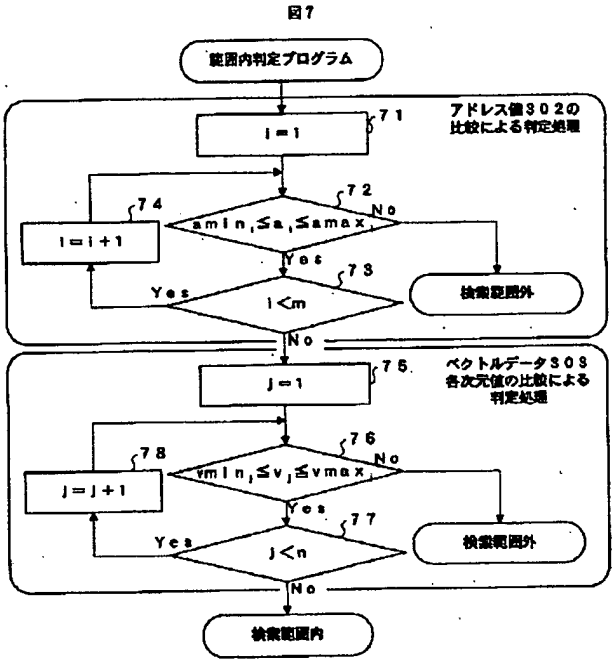
【図６】

図6



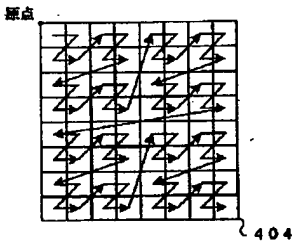
【図７】

図7



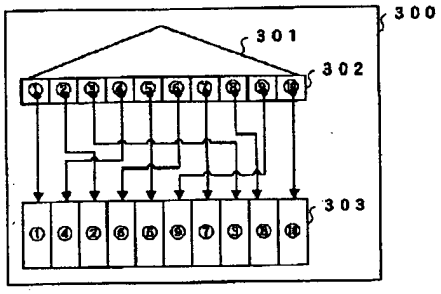
【図９】

図9



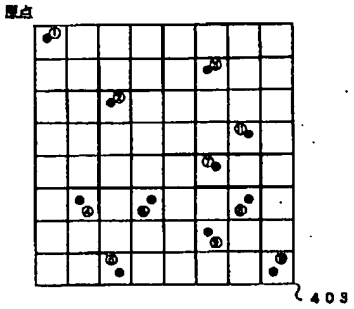
【図１０】

図10



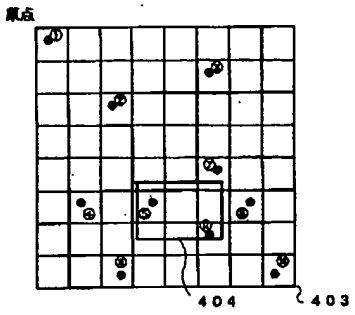
【図１１】

図11



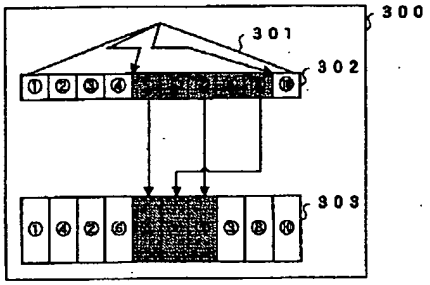
【図１３】

図13



【図１４】

図14



【図１５】

図15

対象データ	判定条件式			判定結果
⑤	横軸	[0.1.1]≦[0.1.1]≦[1.0.1]	○	検索範囲内
	縦軸	[1.0.0]≦[1.0.1]≦[1.1.0]	○	
⑥	横軸	[0.1.1]≦[0.1.0]≦[1.0.1]	×	検索範囲外
	縦軸	[1.0.0]≦[1.1.1]≦[1.1.0]	×	
⑦	横軸	[0.1.1]≦[1.0.1]≦[1.0.1]	○	検索範囲内
	縦軸	[1.0.0]≦[1.0.0]≦[1.1.0]	○	
⑧	横軸	[0.1.1]≦[1.1.0]≦[1.0.1]	×	検索範囲外
	縦軸	[1.0.0]≦[1.0.1]≦[1.1.0]	○	
⑨	横軸	[0.1.1]≦[1.0.0]≦[1.0.1]	○	検索範囲内
	縦軸	[1.0.0]≦[1.1.0]≦[1.1.0]	○	